Primary Vertex resolution after applying beamspot constraint

Diwakar
25/02/19

INFORMATION AVAILABLE IN THE BIG NTUPLES

1. Reconstructed primary vertex
2. Reconstructed primary vertex covariant matrix
3. Beamspot position
4. Beamspot covariant matrix

The idea is to recalculate the primary vertex by using this information by the method of minimising χ^2

RECALCULATING PV BY MINIMISING χ^2

Suppose:

- X(pv) = 3 vector containing PV
- X(bs) = 3 vector containing beamspot position
- V(pv) = 3*3 matrix containing PV covariant matrix (it is symmetric)
- V(bs) = 3*3 matrix containing beamspot covariant matrix (it is also symmetric)

Then χ^2 will be defined as:

$$(x - x(bs))^T V(bs)^{-1} (x - x(bs)) \quad + \quad (x - x(pv))^T V(pv)^{-1} (x - x(pv))$$

The next task is to minimise χ^2 w.r.t. x by taking the derivative of above equation w.r.t. x and putting it equal to 0 (done in next slide). From the resulting equation we will find the value of x for which χ^2 is minimum.

RECALCULATING PV BY MINIMISING χ^2

$$\boxed{\chi^2 = (x - x(bs))^T V(bs)^{-1} (x - x(bs)) + (x - x(pv))^T V(pv)^{-1} (x - x(pv))}$$

x is not a number but a 3 vector or 3*1 matrix.
Now taking derivative w.r.t x and putting it = 0.

$$(x - x(bs))^T V(bs)^{-1} + (x - x(bs))^T (V(bs)^{-1})^T + (x - x(pv))^T V(pv)^{-1} + (x - x(pv))^T (V(pv)^{-1})^T = 0$$

Where we have used the identities:

$$\frac{d(\mathbf{u}^T \mathbf{A} \mathbf{v})}{d\mathbf{x}} = \mathbf{u}^T \mathbf{A} \frac{d\mathbf{v}}{d\mathbf{x}} + \mathbf{v}^T \mathbf{A}^T \frac{d\mathbf{u}}{d\mathbf{x}} \qquad \frac{d\mathbf{x}^T}{d\mathbf{x}} = \mathbf{I}$$

RECALCULATING PV BY MINIMISING χ^2

$$(x - x(bs))^T V(bs)^{-1} + (x - x(bs))^T (V(bs)^{-1})^T + (x - x(pv))^T V(pv)^{-1} + (x - x(pv))^T (V(pv)^{-1})^T = 0$$

Since V(bs) and V(pv) are symmetric, $V(bs)^{-1}$ and $V(pv)^{-1}$ will also be symmetric, i.e.

$$(V(bs)^{-1})^T = V(bs)^{-1} \quad \text{and} \quad (V(pv)^{-1})^T = V(pv)^{-1}$$

Using this result and rearranging terms in above equation, we will get:

$$\boxed{x^T = [x(bs)^T\, 2\, V(bs)^{-1} + x(pv)^T\, 2\, V(pv)^{-1}][2\, V(bs)^{-1} + 2\, V(pv)^{-1}]^{-1}}$$

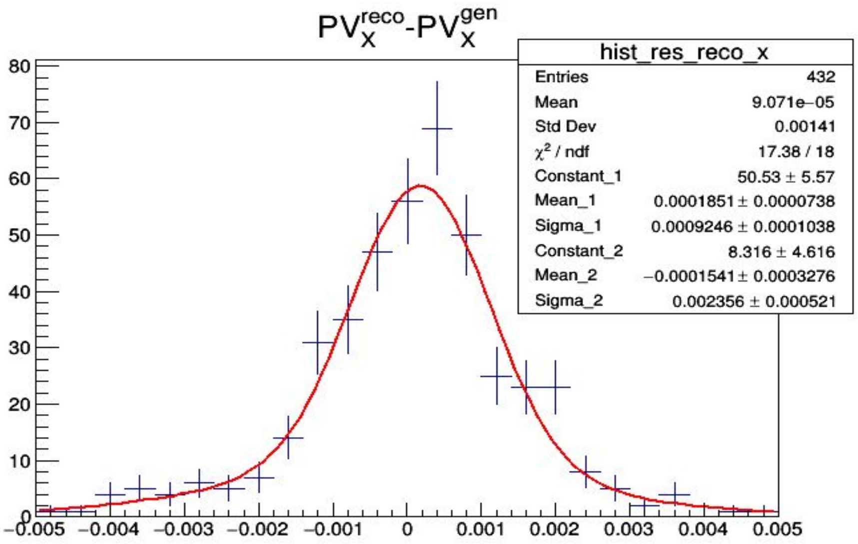
IMPLEMENTATION AND EXECUTION OF CODE

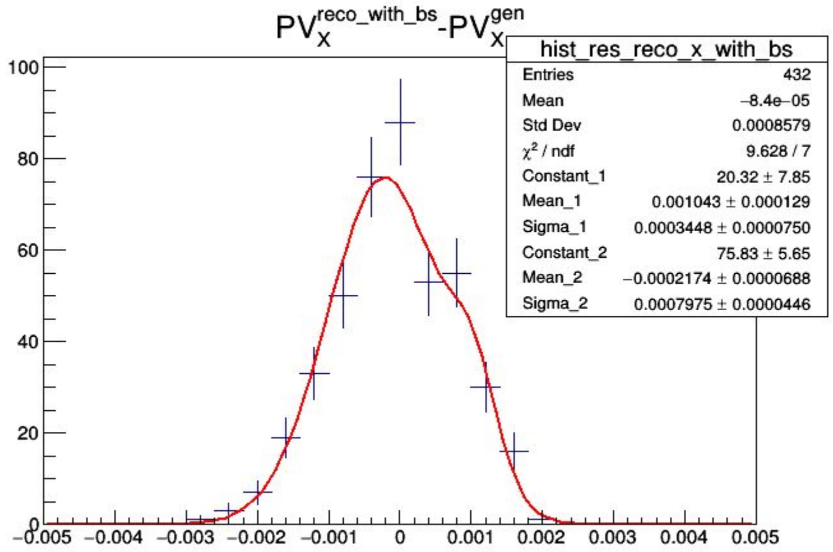
The code was written in SynchNTupleProducer.cpp

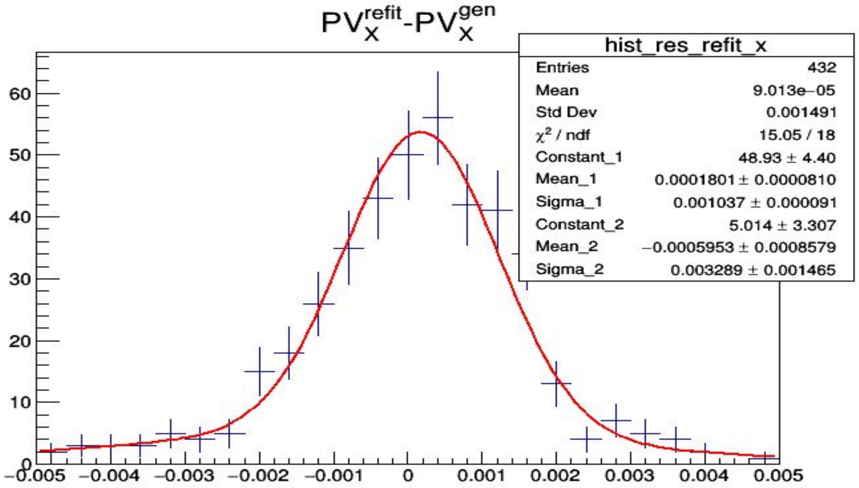
I ran the code for gg->H MC sample.

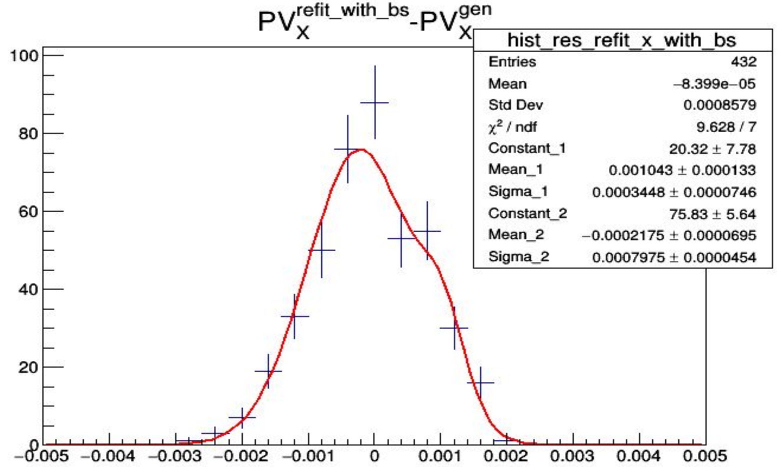
I selected the events in $\mu\tau_h$ channel for producing SynchNTuples.

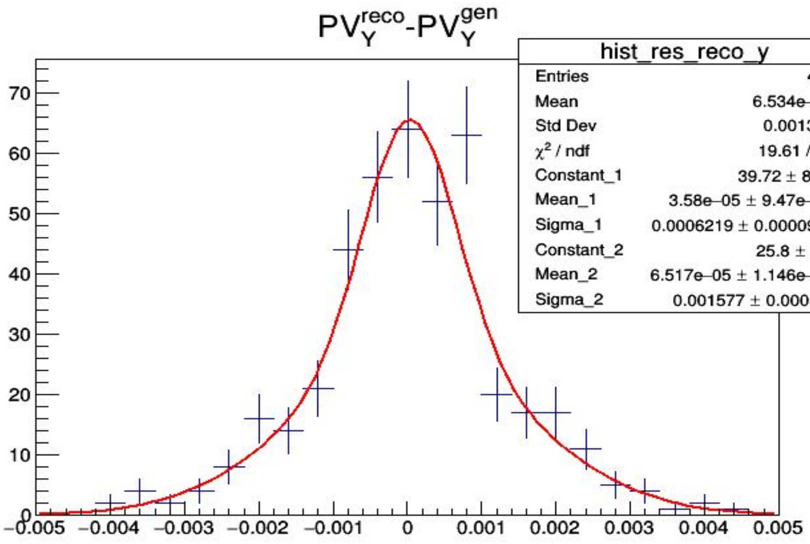
The primary vertex resolution plots are presented in the following slides:

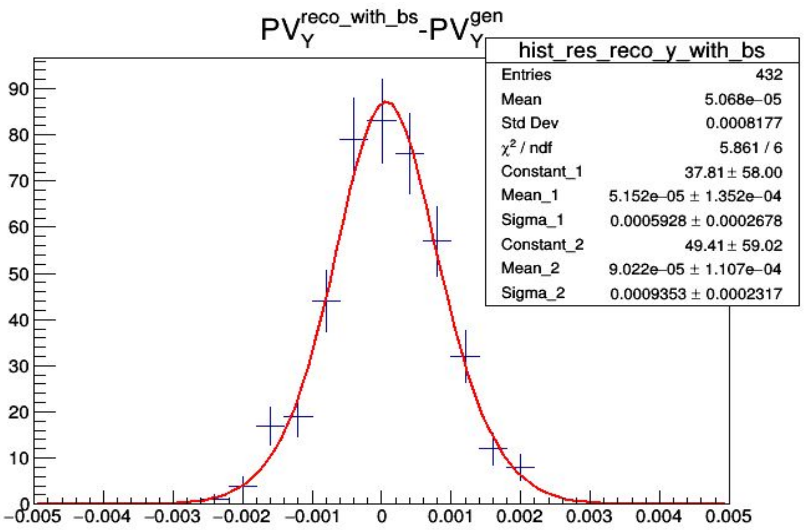

PV_X^{reco}-PV_X^{gen}
hist_res_reco_x
Entries 432
Mean 9.071e−05
Std Dev 0.00141
χ^2 / ndf 17.38 / 18
Constant_1 50.53 ± 5.57
Mean_1 0.0001851 ± 0.0000738
Sigma_1 0.0009246 ± 0.0001038
Constant_2 8.316 ± 4.616
Mean_2 −0.0001541 ± 0.0003276
Sigma_2 0.002356 ± 0.000521

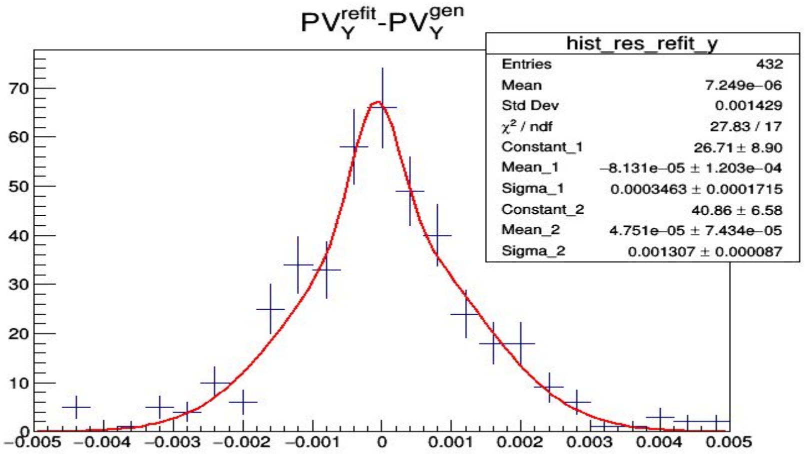

$PV_X^{reco\_with\_bs}$-PV_X^{gen}
hist_res_reco_x_with_bs
Entries 432
Mean −8.4e−05
Std Dev 0.0008579
χ^2 / ndf 9.628 / 7
Constant_1 20.32 ± 7.85
Mean_1 0.001043 ± 0.000129
Sigma_1 0.0003448 ± 0.0000750
Constant_2 75.83 ± 5.65
Mean_2 −0.0002174 ± 0.0000688
Sigma_2 0.0007975 ± 0.0000446

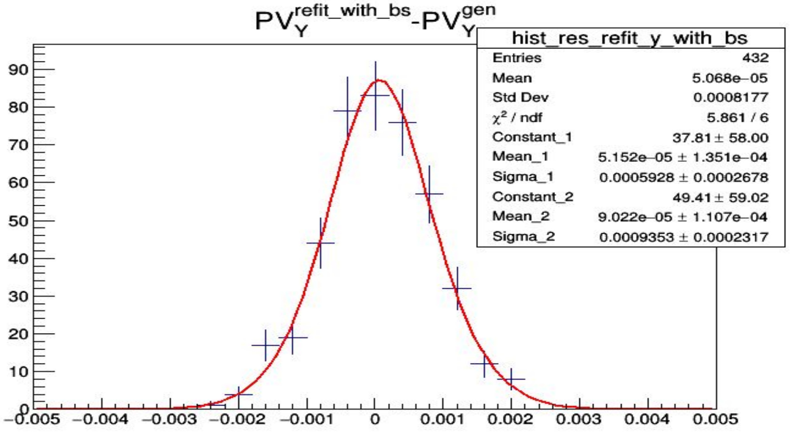

PV_X^{refit}-PV_X^{gen}
hist_res_refit_x
Entries 432
Mean 9.013e−05
Std Dev 0.001491
χ^2 / ndf 15.05 / 18
Constant_1 48.93 ± 4.40
Mean_1 0.0001801 ± 0.0000810
Sigma_1 0.001037 ± 0.000091
Constant_2 5.014 ± 3.307
Mean_2 −0.0005953 ± 0.0008579
Sigma_2 0.003289 ± 0.001465

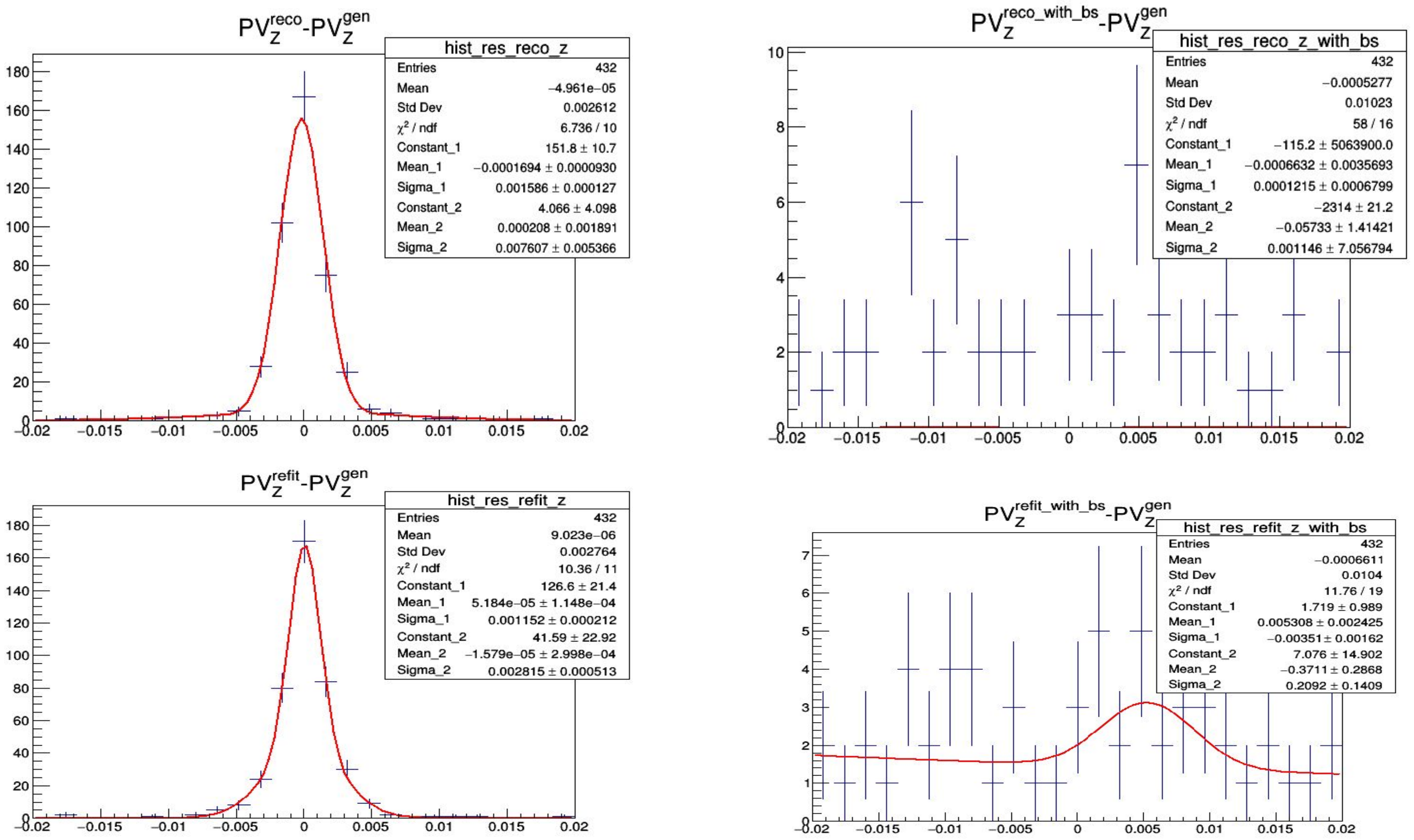

$PV_X^{refit\_with\_bs}$-PV_X^{gen}
hist_res_refit_x_with_bs
Entries 432
Mean −8.399e−05
Std Dev 0.0008579
χ^2 / ndf 9.628 / 7
Constant_1 20.32 ± 7.78
Mean_1 0.001043 ± 0.000133
Sigma_1 0.0003448 ± 0.0000746
Constant_2 75.83 ± 5.64
Mean_2 −0.0002175 ± 0.0000695
Sigma_2 0.0007975 ± 0.0000454


PV_Y^{reco}-PV_Y^{gen}
hist_res_reco_y
Entries 432
Mean 6.534e−05
Std Dev 0.001313
χ² / ndf 19.61 / 16
Constant_1 39.72 ± 8.13
Mean_1 3.58e−05 ± 9.47e−05
Sigma_1 0.0006219 ± 0.0000982
Constant_2 25.8 ± 6.6
Mean_2 6.517e−05 ± 1.146e−04
Sigma_2 0.001577 ± 0.000141


$PV_Y^{reco\_with\_bs}$-PV_Y^{gen}
hist_res_reco_y_with_bs
Entries 432
Mean 5.068e−05
Std Dev 0.0008177
χ² / ndf 5.861 / 6
Constant_1 37.81 ± 58.00
Mean_1 5.152e−05 ± 1.352e−04
Sigma_1 0.0005928 ± 0.0002678
Constant_2 49.41 ± 59.02
Mean_2 9.022e−05 ± 1.107e−04
Sigma_2 0.0009353 ± 0.0002317


PV_Y^{refit}-PV_Y^{gen}
hist_res_refit_y
Entries 432
Mean 7.249e−06
Std Dev 0.001429
χ² / ndf 27.83 / 17
Constant_1 26.71 ± 8.90
Mean_1 −8.131e−05 ± 1.203e−04
Sigma_1 0.0003463 ± 0.0001715
Constant_2 40.86 ± 6.58
Mean_2 4.751e−05 ± 7.434e−05
Sigma_2 0.001307 ± 0.000087


$PV_Y^{refit\_with\_bs}$-PV_Y^{gen}
hist_res_refit_y_with_bs
Entries 432
Mean 5.068e−05
Std Dev 0.0008177
χ² / ndf 5.861 / 6
Constant_1 37.81 ± 58.00
Mean_1 5.152e−05 ± 1.351e−04
Sigma_1 0.0005928 ± 0.0002678
Constant_2 49.41 ± 59.02
Mean_2 9.022e−05 ± 1.107e−04
Sigma_2 0.0009353 ± 0.0002317



PV_Z^reco-PV_Z^gen
hist_res_reco_z
Entries 432
Mean -4.961e-05
Std Dev 0.002612
χ² / ndf 6.736 / 10
Constant_1 151.8 ± 10.7
Mean_1 -0.0001694 ± 0.0000930
Sigma_1 0.001586 ± 0.000127
Constant_2 4.066 ± 4.098
Mean_2 0.000208 ± 0.001891
Sigma_2 0.007607 ± 0.005366
PV_Z^reco_with_bs-PV_Z^gen
hist_res_reco_z_with_bs
Entries 432
Mean -0.0005277
Std Dev 0.01023
χ² / ndf 58 / 16
Constant_1 -115.2 ± 5063900.0
Mean_1 -0.0006632 ± 0.0035693
Sigma_1 0.0001215 ± 0.0006799
Constant_2 -2314 ± 21.2
Mean_2 -0.05733 ± 1.41421
Sigma_2 0.001146 ± 7.056794
PV_Z^refit-PV_Z^gen
hist_res_refit_z
Entries 432
Mean 9.023e-06
Std Dev 0.002764
χ² / ndf 10.36 / 11
Constant_1 126.6 ± 21.4
Mean_1 5.184e-05 ± 1.148e-04
Sigma_1 0.001152 ± 0.000212
Constant_2 41.59 ± 22.92
Mean_2 -1.579e-05 ± 2.998e-04
Sigma_2 0.002815 ± 0.000513
PV_Z^refit_with_bs-PV_Z^gen
hist_res_refit_z_with_bs
Entries 432
Mean -0.0006611
Std Dev 0.0104
χ² / ndf 11.76 / 19
Constant_1 1.719 ± 0.989
Mean_1 0.005308 ± 0.002425
Sigma_1 -0.00351 ± 0.00162
Constant_2 7.076 ± 14.902
Mean_2 -0.3711 ± 0.2868
Sigma_2 0.2092 ± 0.1409
-0.02 -0.015 -0.01 -0.005 0 0.005 0.01 0.015 0.02

CONCLUSIONS

- Beamspot constraint leads to a big improvement in resolution in x and y direction but it miscalculates the z direction totally.
- The effect of refitting is being totally washed out upon applying beamspot constraint in x and y direction.

Primary Vertex resolution after applying beamspot constraint (cont.)

Diwakar

11 Mar 2019

Done previously:

- Applied beamspot constraint to the primary vertex (both reconstructed and refitted) and checked the improvement in resolution.
- Resolution was checked in all three directions : x,y,z
- Big improvement was seen in x and y directions (SD decreased by a factor of 10 after applying BS)
- z direction was getting miscalculated totally.
- “Refitted vertex with BS” value was being calculated almost equal to the value for “Reconstructed vertex with BS”.

Further progress:

- Used more events (~4000 compared to last time’s ~400)
- Dropped the z direction totally i.e. beamspot is now being applied only in x and y direction and the covariant matrices are 2*2 instead of 3*3.
- Found the reason why value of “Refitted vertex with BS” was being calculated almost equal to the value for “Reconstructed vertex with BS”.

A few observations:

- I had been using gg->H->tautau MC file and noticed a few things:
- All the events (about 93000) in that file have same value of “beamspot_position” and is equal to (x,y) = (-0.0247936, 0.0692861).
- Also all the events have same value of beamspot covariant matrix equal to :

$$\begin{bmatrix} 3.0575 * 10^{-11} & -4.57847 * 10^{-14} \\ -4.57847 * 10^{-14} & 3.09172 * 10^{-11} \end{bmatrix}$$

A few observations (cont.)

- The reconstructed PV covariant matrix is much bigger than beamspot covariant matrix:

$$\begin{bmatrix} \sim O(10^{-11}) & \sim O(10^{-14}) \\ \sim O(10^{-14}) & \sim O(10^{-11}) \end{bmatrix}$$

BEAMSPOT

BIGGER

$$\begin{bmatrix} \sim O(10^{-6}) & \sim O(10^{-7}) \\ \sim O(10^{-7}) & \sim O(10^{-6}) \end{bmatrix}$$

RECONSTRUCTED PV

- Consequently beamspot covariant inverse is much bigger:

$$\begin{bmatrix} \sim O(10^{11}) & \sim O(10^{8}) \\ \sim O(10^{8}) & \sim O(10^{11}) \end{bmatrix}$$

INVERSE BEAMSPOT

BIGGER

$$\begin{bmatrix} \sim O(10^{6}) & \sim O(10^{5}) \\ \sim O(10^{5}) & \sim O(10^{6}) \end{bmatrix}$$

INVERSE RECONSTRUCTED PV

A look at the formula of beamspot constraint

$$x^T = [\, x(bs)^T\, V(bs)^{-1} + x(pv)^T\, V(pv)^{-1} \,][\, V(bs)^{-1} + V(pv)^{-1} \,]^{-1}$$

where $V(bs)^{-1}$ = inverse of beamspot cov matrix

$V(pv)^{-1}$ = inverse of reconstructed pv cov matrix

$V(bs)^{-1} >> V(pv)^{-1}$

$$x^T = [\, x(bs)^T\, V(bs)^{-1} + x(pv)^T\, V(pv)^{-1} \,][\, V(bs)^{-1} + V(pv)^{-1} \,]^{-1}$$

SMALL

$$\boxed{x \approx x(bs)}$$

$$\boxed{x \approx x(bs)}$$

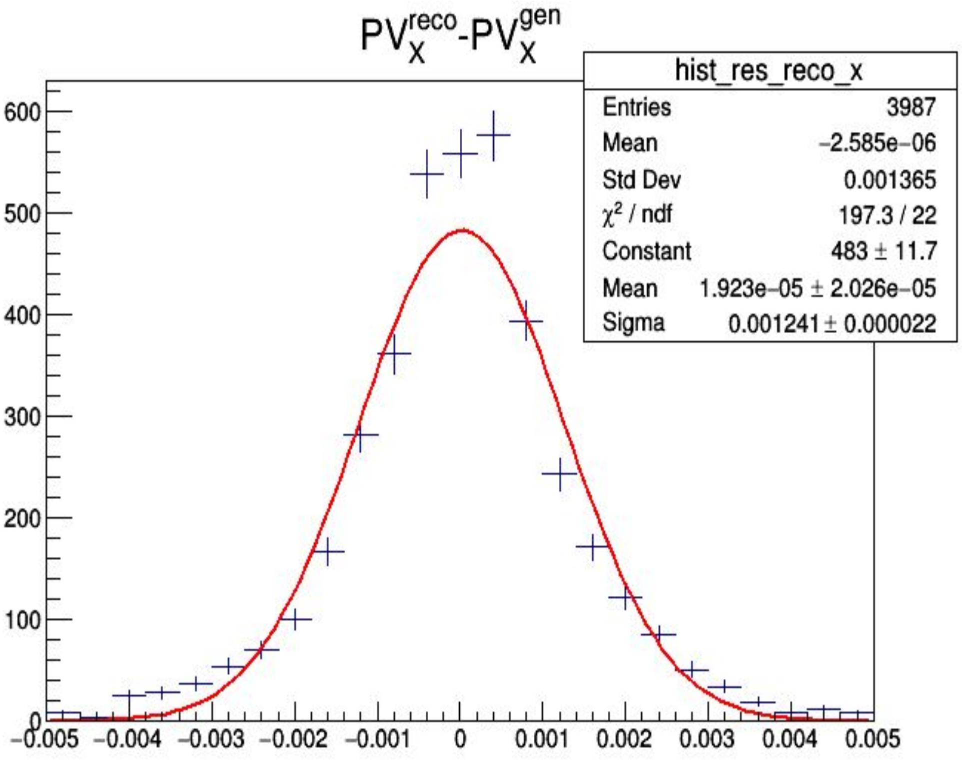
The value of recalculated PV that we get is actually very close to the beamspot position (the difference is seen in 6th or 7th place in decimal).

This is actually the reason why last time i was getting almost same value of reconstructed and refitted PV after applying beamspot constraint (i was using reconstructed cov matrix for refitted recalculation as well); they were both coming almost equal to x(bs).

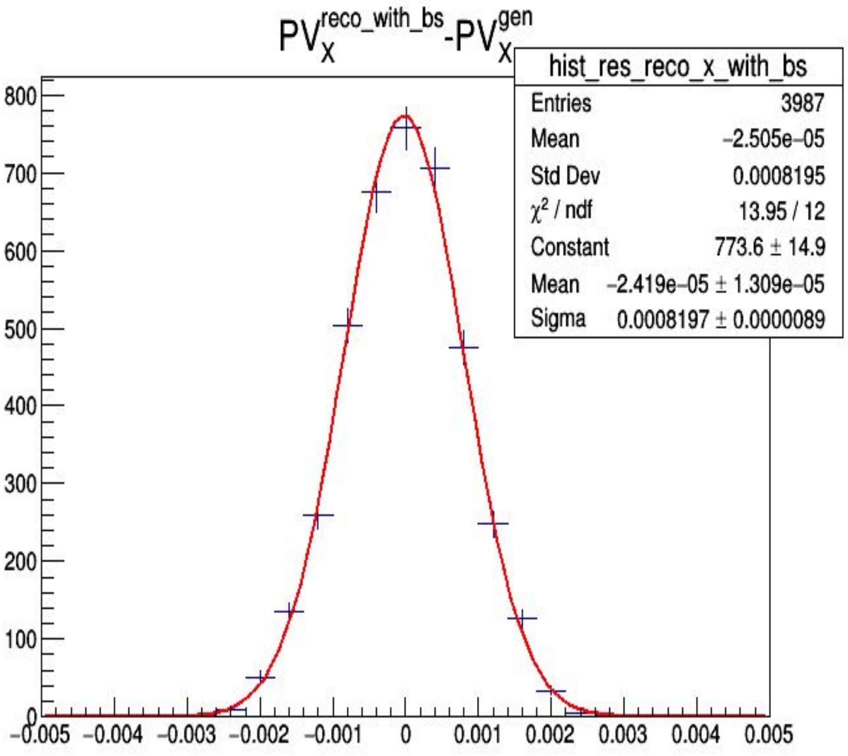
Plots

- More statistics were used this time (~4000 compared to ~400 last time)
- The resolution was found by fitting single gaussian.
- The beamspot constraint could not be applied to refitted vertices because their covariant matrices were equal to zero matrices (this could be a bug)

Plots in x direction

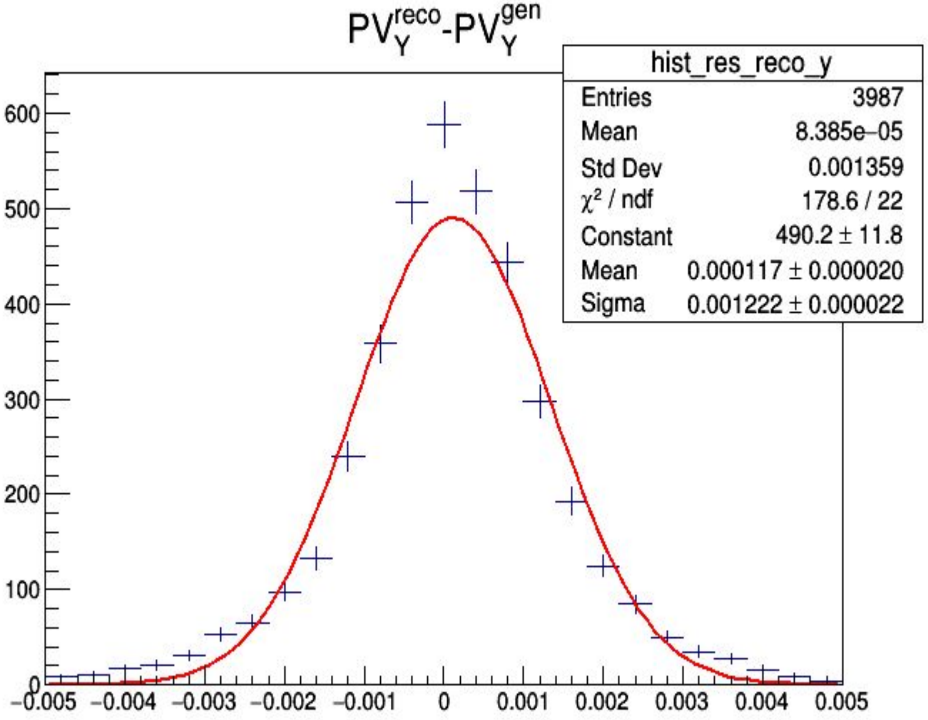


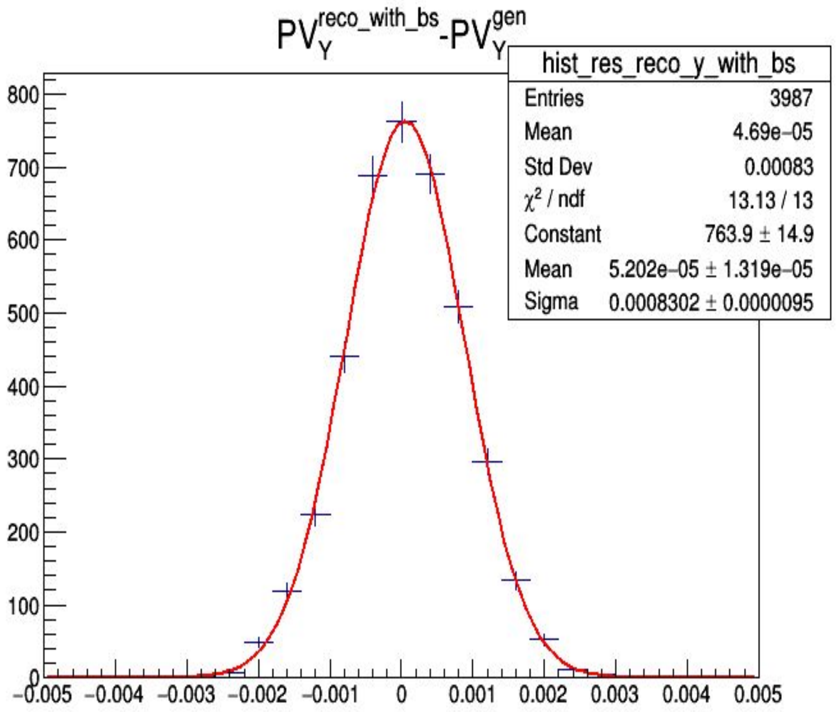
SD = 0.001241



SD = 0.0008197

Plots in y direction





SD = 0.001222

SD = 0.0008302

Next steps...

- Use more statistics by including all the events available in the file. Till now i have been using the mt channel in Synch ntuple. I can use all the events in the big ntuple to do the studies.
- I plan to check the effect of bs constraint on the resolution of PCA and IP.

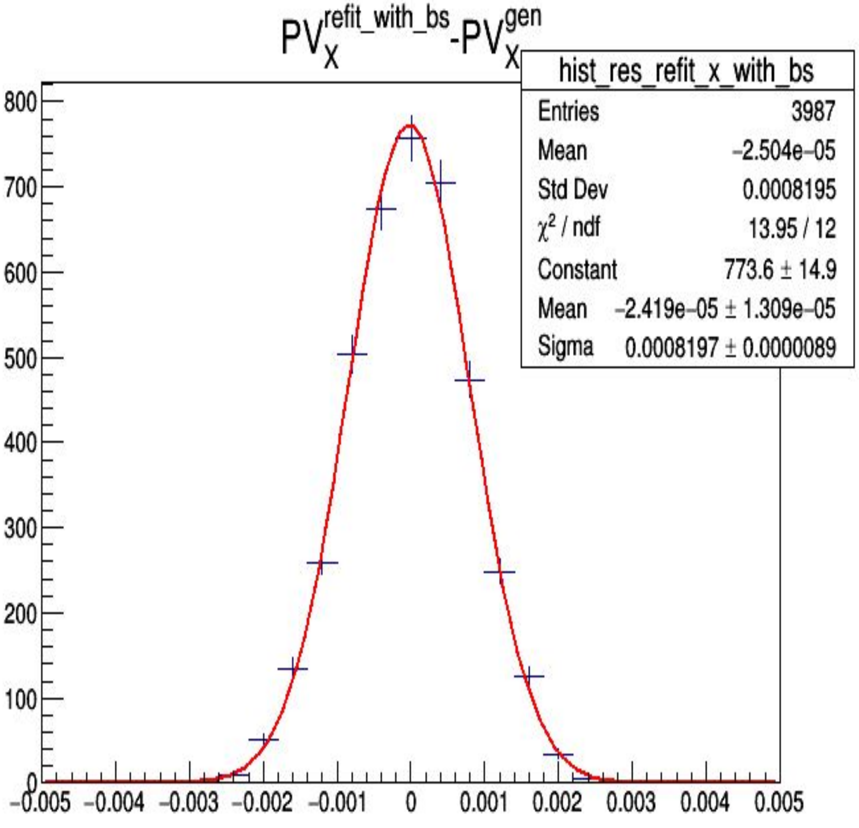
Primary Vertex resolution after applying beamspot constraint (cont.)

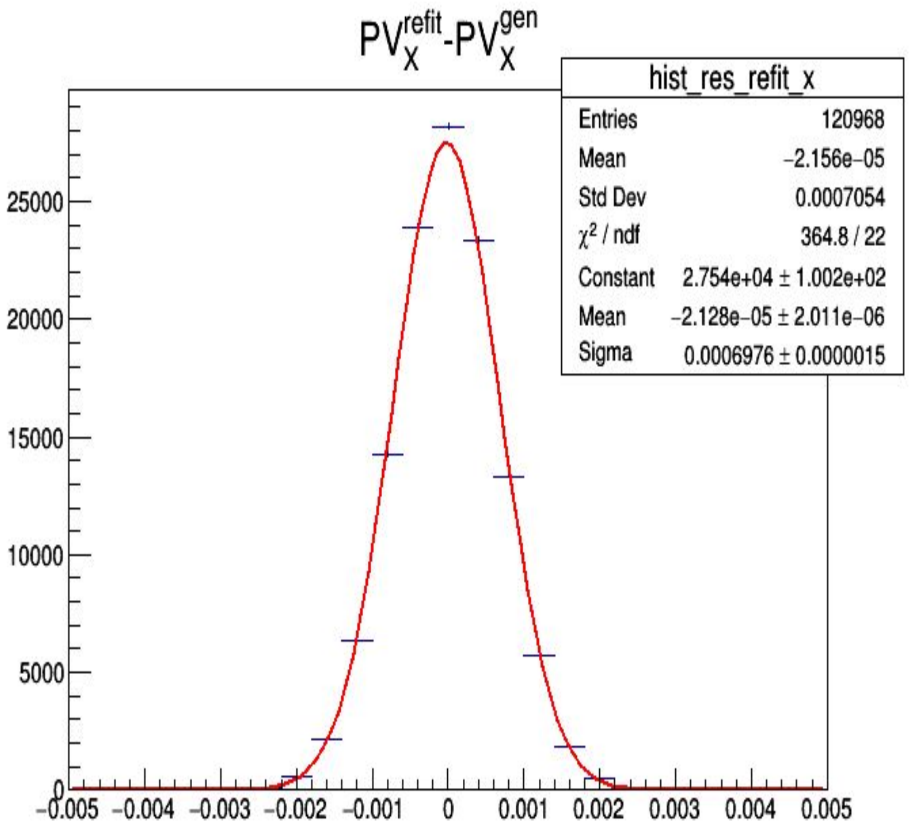
Diwakar

18 Mar 2019

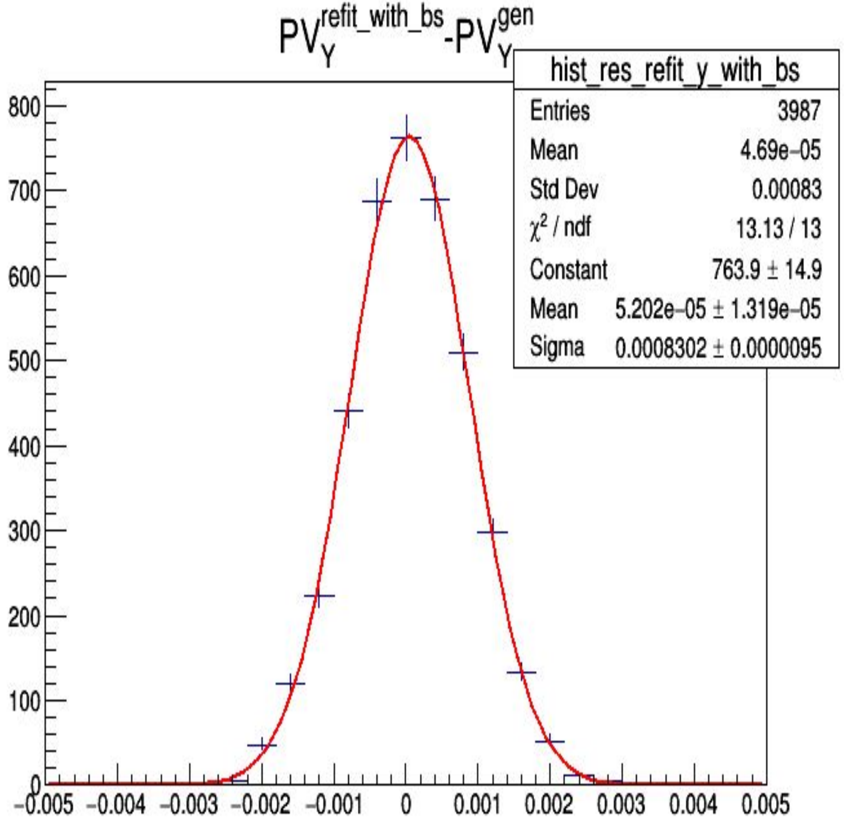
1. Corrected the bug that was producing null covariant refitted matrices
2. Made a comparison between the offline beamspot constraint results and the results produced by applying beamspot constraint using the tools in CMSSW

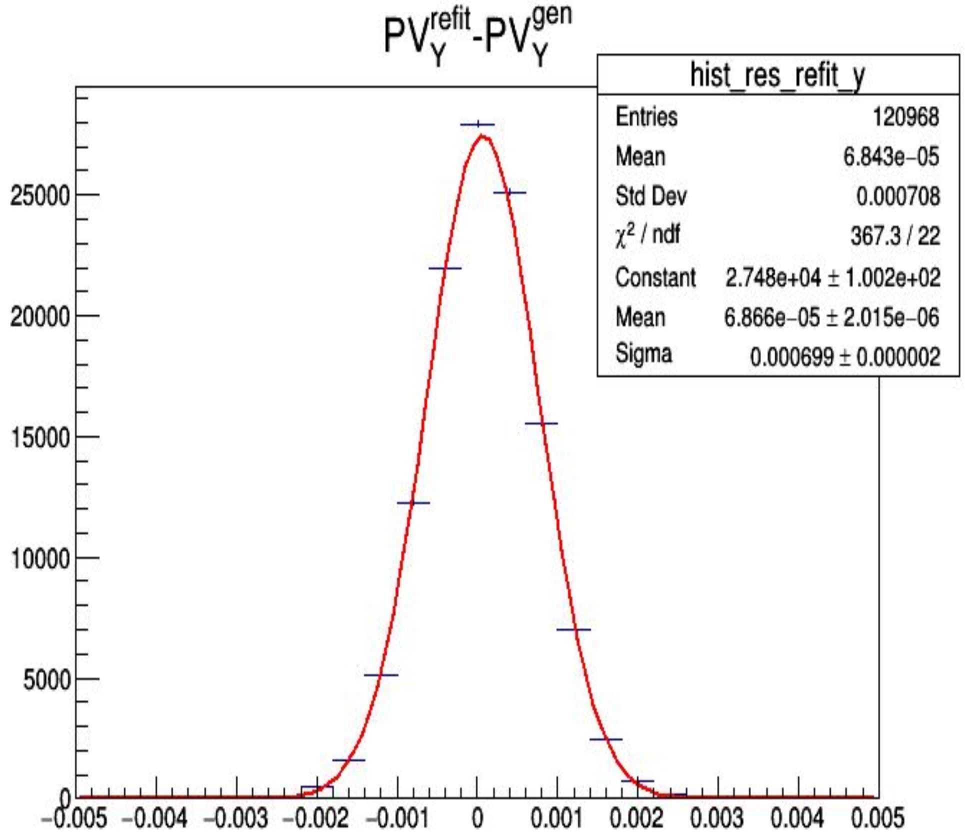
OLD SAMPLE (OFFLINE BS CONSTRAINT) | **NEW SAMPLE (CMSSW BS CONSTRAINT)**





OLD SAMPLE (OFFLINE BS CONSTRAINT) **NEW SAMPLE (CMSSW BS CONSTRAINT)**





Primary Vertex resolution after applying beamspot constraint (cont.)

Diwakar

1 April 2019

SHOWN LAST TIME:

1. Comparison of PV resolution with BS constraint in two cases:

   BS constraint was applied offline (by method of minimising chi2)

   BS constraint was applied online (using CMSSW in built functions)

2. Resolution was found to be better in 2nd case (online). Also the new PV was not equal to beamspot position in 2nd case.

FOUR CHOICES

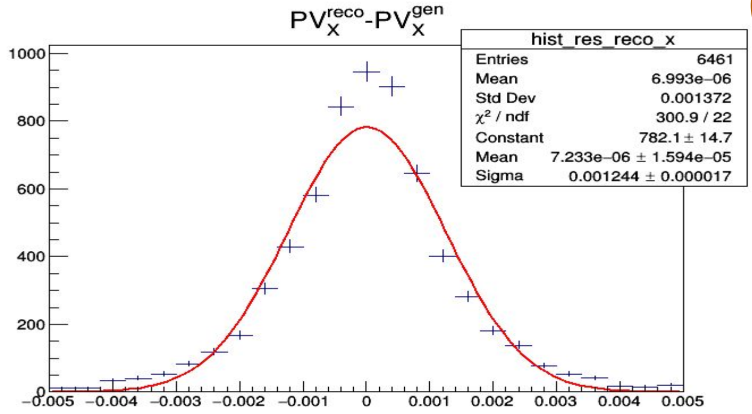
- Primary vertex
- Primary vertex with BS constraint
- Refitted Primary vertex
- Refitted Primary vertex with BS constraint

HOW TO CHOOSE

- Compare resolution of PV (in x, y, z)
- Compare resolution of PCA (in x, y, z)
- Compare resolution of IP

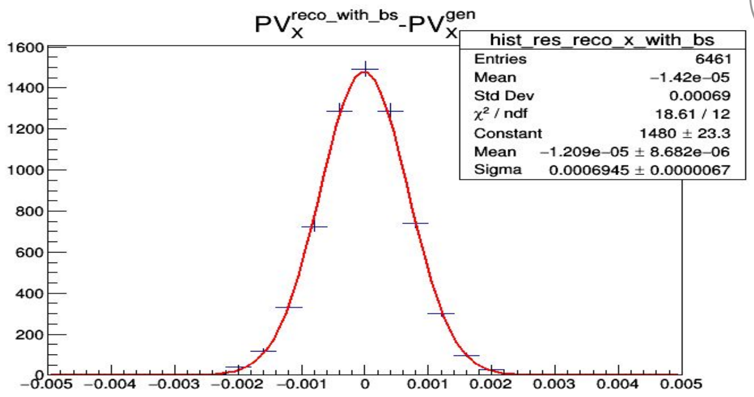
PRIMARY VERTEX

3



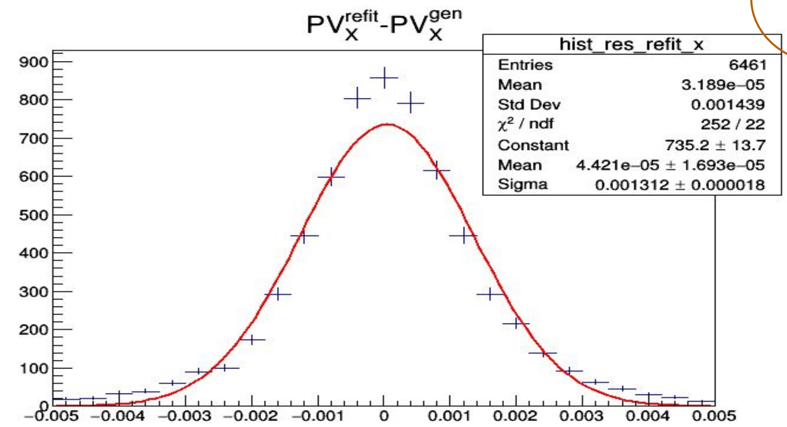
PRIMARY VERTEX WITH BS

2



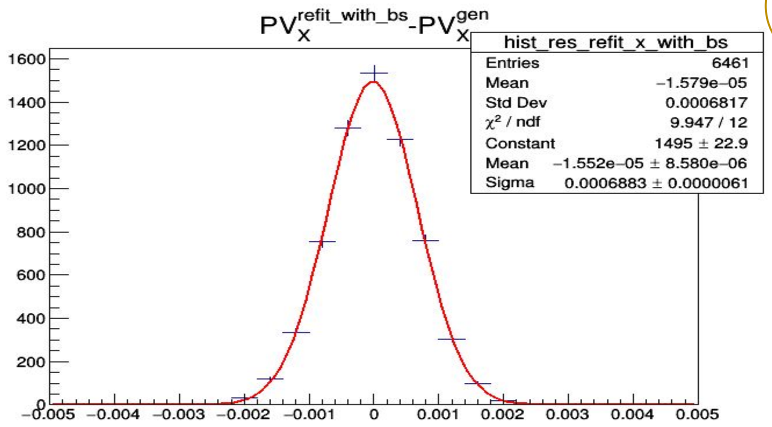
REFITTED PRIMARY VERTEX

4



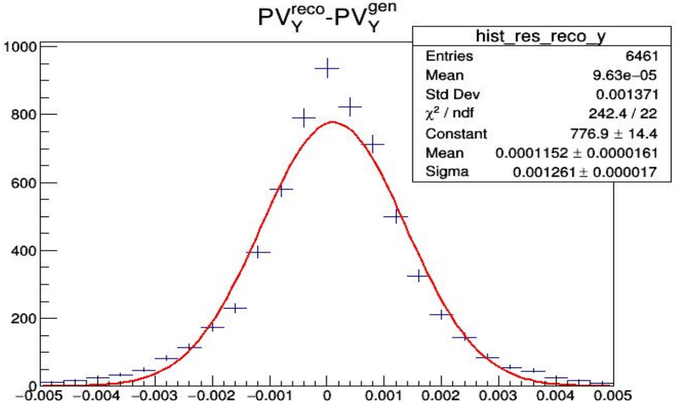
REFITTED PRIMARY VERTEX WITH BS

1



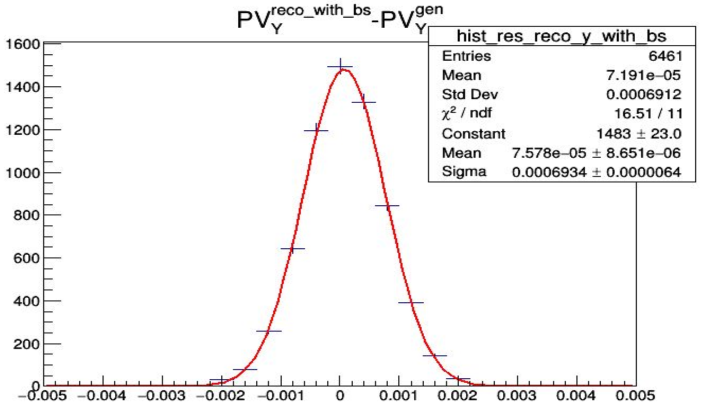
PRIMARY VERTEX

3



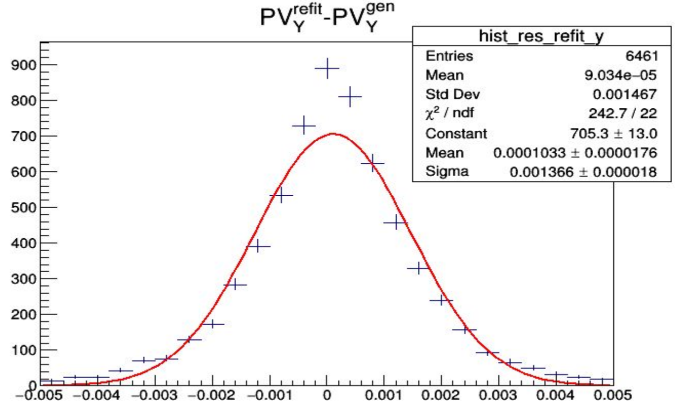
PRIMARY VERTEX WITH BS

1



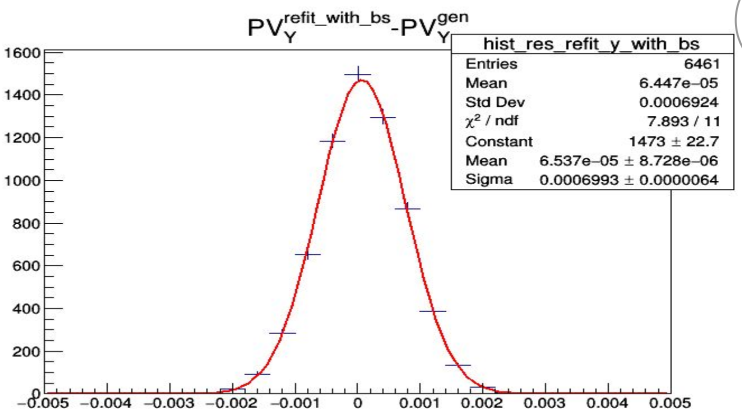
REFITTED PRIMARY VERTEX

4



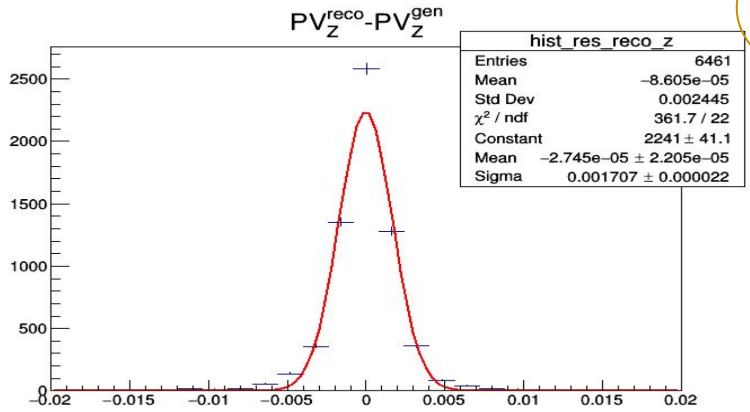
REFITTED PRIMARY VERTEX WITH BS

2



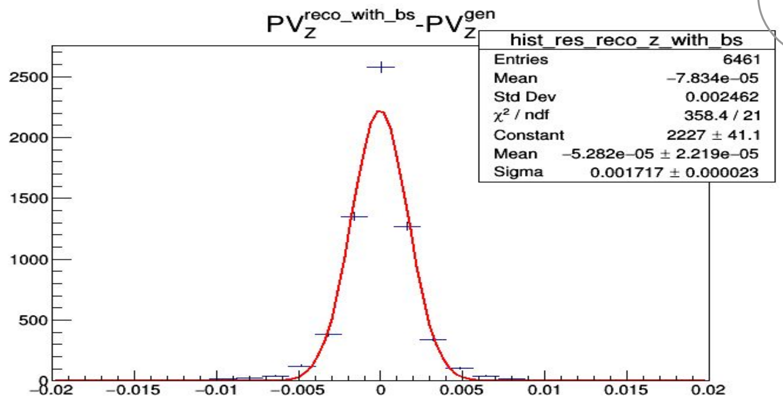
PRIMARY VERTEX

1



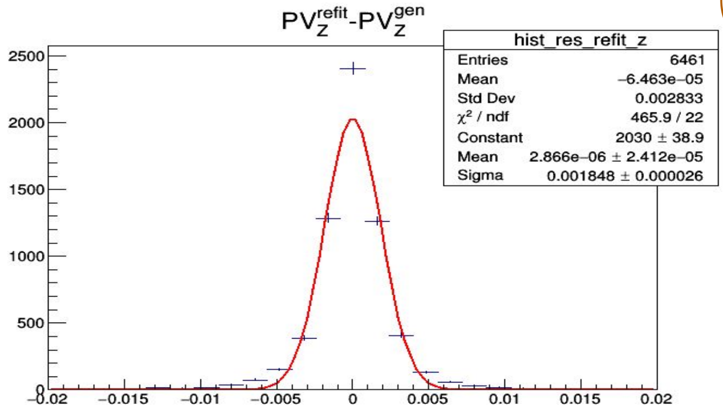
PRIMARY VERTEX WITH BS

2



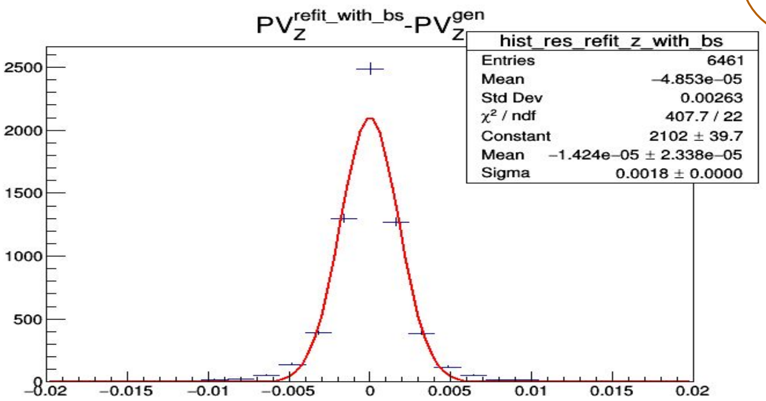
REFITTED PRIMARY VERTEX

4



REFITTED PRIMARY VERTEX WITH BS

3



RESOLUTION TABLE

	PRIMARY VERTEX	PRIMARY VERTEX WITH BS	REFITTED PRIMARY VERTEX	REFITTED PRIMARY VERTEX WITH BS
X	0.001244	0.0006945	0.001312	0.0006883
Y	0.001261	0.0006934	0.001366	0.0006993
Z	0.001707	0.001717	0.001848	0.0018

CONCLUSIONS

- Using Primary vertex with BS constraint seems to be the best option.

- For applying BS constraint, using CMSSW algorithm is better than doing it offline.